

# A Las Vegas algorithm to solve the elliptic curve discrete logarithm problem

Ayan Mahalanobis\*

Vivek Mallick†

IISER Pune, Pashan, Pune 411008, India

## Abstract

In this short paper, we develop a probabilistic algorithm for the elliptic curve discrete logarithm problem. This algorithm is not generic in nature, it uses some properties of the elliptic curve.

## 1 Introduction

Public-key cryptography is a backbone of this modern society. Many of the public-key cryptosystems depend on the *discrete logarithm problem* as their cryptographic primitive. Of all the groups used in a discrete logarithm based protocol, the group of *rational points of an elliptic curve* is the most popular. In this paper, we describe an attack on the elliptic curve discrete logarithm problem.

There are two kinds of attack on the discrete logarithm problem. One is generic. This kind of attack works in any group. Examples of such attacks are the baby-step giant-step attack [3, Proposition 2.22] and Pollard's rho [3, Section 4.5]. The other kind of attack depends on the group used. Example of such attack is the index-calculus attack [3, Section 3.8] on the multiplicative group of a finite field.

In this paper, we describe an attack which is particular to the elliptic curves. The attack uses some properties of the elliptic curve (Theorem 3.1). The attack is a Las Vegas algorithm. We describe the probability of success for this algorithm.

At the end, we present three algorithms to solve the discrete logarithm problem. We could have presented only the last algorithm instead, because it sub assumes the ones before it. However, this presentation preserves the way this project progressed over the years and benefits the exposition of ideas.

---

\*ayan.mahalanobis@gmail.com. Research partially supported by a SERB and NBHM research grants.

†vmallick@iiserpune.ac.in

## 2 The central idea behind our attack

Let  $G$  be a cyclic group of prime order  $p$ . Let  $P$  be a non-identity element and  $Q(= mP)$  belong to  $G$ . The *discrete logarithm problem* is to compute the  $m$ . One way to find  $m$  is to find integers  $n_i$ , for  $i = 1, 2, \dots, k$  for some positive integer  $k$  and  $1 \leq n_i < p$  such that  $\sum_{i=1}^k n_i = m$ . The last equality is hard to compute because we do not know  $m$ . However we can decide whether

$$\sum_{i=1}^k n_i P = Q \quad (1)$$

and based on that we can decide if  $\sum_{i=1}^k n_i = m$ . Once the equality holds, we have found  $m$  and the discrete logarithm problem is solved.

The number of possible choices of  $n_i$  for a given  $k$  that can solve the discrete logarithm problem is the number of partitions of  $m$  into  $k$  parts. The applicability of the above method depends on, how fast can one decide on the equality in the above equation and on the probability, how likely is it that a given set of positive integers  $n_i$  forms a partition of  $m$ ? An obvious question is raised, can one choose a set of  $n_i$  in such a way that the probability of an equality is higher than random selection? In the next section, we find a way to check for equality in the case of elliptic curves, however our choice of  $n_i$  is uniformly random. Then the algorithm is somewhat straightforward, fix a  $k$ , choose  $n_i$  uniformly random and then check for equality. Once there is a set of  $n_i$  for which the equality is found, we have solved the discrete logarithm problem.

### 2.1 A bit about partitions

Given a positive integer  $m$ , a partition of  $m$  into  $k$  parts is a set of positive integers  $\{n_1, n_2, \dots, n_k\}$  such that  $\sum_{i=1}^k n_i = m$ . Number of such partitions is denoted by  $p(m, k)$ . It is customary to assume that  $p(1, k) = 0$  for all positive  $k$ . Note that while defining partitions, we use the notion of a set of integers, so the order of appearance of the parts  $n_i$  do not matter. When the parts are distinct, we call it a unique partition and denote its number by  $q(m, k)$ . The theory of partitions was intensely studied by many great mathematicians. Our interest in the theory of partitions is to find a good and nice approximation to  $q(m, k)$  for a fixed  $k$ . For this we are going to use the work of Knessl and Keller [4]. Using recursion, they found a suitable approximation both for  $p(m, k)$  and  $q(m, k)$ . They further show that for fixed  $k$  and large enough  $m$ ,  $p(m, k)$  and  $q(m, k)$  are approximately equal and an approximation for both is

$$\frac{m^{k-1}}{k[(k-1)!]^2}. \quad (2)$$

We will use this estimate to estimate the probability of success of our Las Vegas algorithm.

### 3 Elliptic curve discrete logarithm problem

The elliptic curve discrete logarithm problem (ECDLP) is the heart and soul of modern public-key cryptography. This paper is about a new probabilistic algorithm to solve this problem. Our algorithm is a fairly straightforward application of the Riemann-Roch theorem. We denote by  $\mathcal{E}(\mathbb{F}_q)$  the group of rational points of the elliptic curve  $\mathcal{E}$  over  $\mathbb{F}_q$ . It is well known that there is an isomorphism  $\mathcal{E}(\mathbb{F}_q) \rightarrow \text{Pic}^0(\mathcal{E})$  given by  $P \mapsto [P] - [\mathcal{O}]$  [5, Proposition 4.10].

**Theorem 3.1.** *Let  $\mathcal{E}$  be an elliptic curve over  $\mathbb{F}_q$  and  $P_1, P_2, \dots, P_k$  be points on that curve, where  $k = 3n'$  for some positive integer  $n'$ . Then  $\sum_{i=1}^k P_i = \mathcal{O}$  if and only if there is a curve  $\mathcal{C}$  of degree  $n'$  that passes through these points. Multiplicities are intersection multiplicities.*

*Proof.* Assume that  $\sum_{i=1}^k P_i = \mathcal{O}$  in  $\mathbb{F}_q$  and then it is such in the algebraic closure  $\bar{\mathbb{F}}_q$ . From the above isomorphism,  $\sum_{i=1}^k P_i \mapsto \sum_{i=1}^k [P_i] - k[\mathcal{O}]$ . Then  $\sum_{i=1}^k [P_i] - k[\mathcal{O}]$  is zero in the Picard group  $\text{Pic}_{\bar{\mathbb{F}}_q}^0(\mathcal{E})$ . Then there is a rational function  $\frac{\phi}{z^{n'}}$  over  $\bar{\mathbb{F}}_q$  such that

$$\sum_{i=1}^k [P_i] - k[\mathcal{O}] = \text{div} \left( \frac{\phi}{z^{n'}} \right) \quad (3)$$

Bezout's theorem justifies that  $\deg(\phi) = n'$ , since  $\phi$  is zero on  $P_1, P_2, \dots, P_k$ . We now claim, there is  $\psi$  over  $\mathbb{F}_q$  which is also of degree  $n'$  and passes through  $P_1, P_2, \dots, P_k$ . First thing to note is that there is a finite extension of  $\mathbb{F}_q$ ,  $\mathbb{F}_{q^N}$  (say) in which all the coefficients of  $\phi$  lies and  $\gcd(q, N) = 1$ . Let  $\mathcal{G}$  be the Galois group of  $\mathbb{F}_{q^N}$  over  $\mathbb{F}_q$  and define

$$\psi = \sum_{\sigma \in \mathcal{G}} \phi^\sigma. \quad (4)$$

Clearly  $\deg(\psi) = n'$ . Note that, since  $P_i$  for  $i = 1, 2, \dots, k$  is in  $\mathbb{F}_q$  is invariant under  $\sigma$ . Furthermore,  $\sigma$  being a field automorphism,  $P_i$  is a zero of  $\phi^\sigma$  for all  $\sigma \in \mathcal{G}$ . This proves that  $P_i$  are zeros of  $\psi$  and then Bezout's theorem shows that these are the all possible zeros of  $\psi$  on  $\mathcal{E}$ . The only thing left to show is that  $\psi$  is over  $\mathbb{F}_q$ . To see that, let's write  $\phi = \sum_{i+j+k=n'} a_{ijk} x^i y^j z^k$ . Then  $\psi = \sum_{i+j+k=n'} \sum_{\sigma \in \mathcal{G}} a_{ijk}^\sigma x^i y^j z^k$ . However, it is well known that  $\sum_{\sigma \in \mathcal{G}} a^\sigma \in \mathbb{F}_q$  for all  $a \in \mathbb{F}_{q^N}$ .

Conversely, if we are given a curve  $\mathcal{C}$  of degree  $n'$  that passes through  $P_1, P_2, \dots, P_k$ . Then consider the rational function  $\mathcal{C}/z^{n'}$ . Then this function has zeros on  $P_i$ ,  $i = 1, 2, \dots, k$  and poles of order  $k$  at  $\mathcal{O}$ . Then the above isomorphism says that  $\sum_{i=1}^k P_i = \mathcal{O}$ .  $\square$

### 3.1 How to use the above theorem in our algorithm

We choose  $k$  such that  $k + 1 = 3n'$  for some positive integer  $n'$ . Then we choose  $k$  random points  $P_1, P_2, \dots, P_k$  from  $\mathcal{E}$  and check if there is a homogeneous curve of degree  $n'$  that passes through these  $k$  points and  $-Q$ . If there is a curve, then the discrete logarithm problem is solved. Otherwise repeat the process by choosing a new set of points  $P_1, P_2, \dots, P_k$ . To choose these points  $P_i$ , we choose a random point  $n_i$  and compute  $n_i P$ . For practical reasons explained later, we would choose  $n_i$  to be distinct from the ones chosen before. This gives rise to distinct points  $P_i$  on  $\mathcal{E}$ .

The only question remains, how do we say if there is a homogeneous curve of degree  $n'$  passing through these selected points? One can answer this question using linear algebra.

Let  $C = \sum_{i+j+k=n'} a_{ijk} x^i y^j z^k$  be a *complete* homogeneous curve of degree  $n'$ . We assume that an ordering of  $i, j, k$  is fixed throughout this paper and  $C$  is presented according to that ordering. By complete we mean that the curve has all the possible monomials of degree  $n'$ . We need to check if  $P_i$ ,  $i = 1, 2, \dots, k$  and  $-Q$  satisfy the curve  $C$ . Note that, there is no need to compute the values of  $a_{ijk}$ , just mere existence will solve the discrete logarithm problem.

Let  $P$  be a point on  $\mathcal{E}$ . We denote by  $\overline{P}$  the value of  $C$  when the values of  $x, y, z$  in  $P$  is substituted in  $C$ . In other words,  $\overline{P}$  is a linear combination of  $a_{ijk}$  with the fixed ordering. We now form a matrix  $\mathcal{M}$  where the rows of  $\mathcal{M}$  are  $\overline{P_i}$  for  $i = 1, 2, \dots, k$  and the last row is  $\overline{-Q}$ . If this matrix has a non-zero left-kernel, we have solved the discrete logarithm problem. By *left-kernel* we mean the kernel of  $\mathcal{M}^T$ , the transpose of  $\mathcal{M}$ .

### 3.2 Why look at the left-kernel instead of the kernel

In this paper, we will use the left-kernel more often than the (right)kernel of  $\mathcal{M}$ . We denote the left-kernel by  $\mathcal{K}$  and kernel by  $\mathcal{K}'$ . We first prove the following theorem:

**Theorem 3.2.** *The following are equivalent:*

- (a)  $\mathcal{K} = 0$ .
- (b)  $\mathcal{K}'$  only contain curves that are a multiple of  $\mathcal{E}$ .

*Proof.* The proof uses a simple counting argument. First recall the well-known fact that the number of monomials of degree  $d$  is  $\binom{d}{2}$ . Furthermore,

notice two things – all multiples of  $\mathcal{E}$  belongs to  $\mathcal{K}'$  and the dimension of that vector-space (multiples of  $\mathcal{E}$ ) is  $\binom{n'-3}{2} = \frac{(n'-2)(n'-1)}{2}$ , where  $n'$  is as defined earlier.

Now,  $\mathcal{M}$  was as defined earlier, has  $3n'$  rows and  $\frac{(n'+1)(n'+2)}{2}$  columns. Then  $\mathcal{K} = 0$  means that the row-rank of  $\mathcal{M}$  is  $3n'$ . So the dimension of the  $\mathcal{K}'$  is

$$\frac{(n'+1)(n'+2)}{2} - 3n' = \frac{(n'-2)(n'-1)}{2}.$$

This proves (a) implies (b).

Conversely, if  $\mathcal{K}'$  contains all the curves that are a multiple of  $\mathcal{E}$  then its dimension is at least  $\frac{(n'-2)(n'-1)}{2}$ , then the rank is  $3n'$ , making  $\mathcal{K} = 0$ .  $\square$

It is easy to see that, while working with the above theorem  $\mathcal{M}$  cannot repeat any row. So from now onwards we would assume that  $\mathcal{M}$  has no repeating rows. For all practical purposes this means that we are working with distinct(unique) partitions.

A question that becomes significantly important later is, instead of choosing  $k+1$  points from the elliptic curve what happens if we choose  $k+l$  points for some positive integer  $l$ . The answer to the question lies in the following corollary.

**Lemma 3.1.** *If  $l \geq 1$ , the dimension of the left kernel of  $\mathcal{M}$  is  $l$ .*

*Proof.* First assume  $l \geq 1$ . In this case, any non-trivial element of  $\mathcal{K}'$  will define a curve which passes through more than  $3d$  point of the elliptic curve. Since the elliptic curve is irreducible, it must be a component of the curve. Thus the equation defining the curve must be divisible by the equation defining the elliptic curve. Thus, the dimension of  $\mathcal{K}'$  is the dimension of all degree  $d$  homogeneous polynomials which are divisible by the elliptic curve. This is the same as the dimension of all degree  $d-3$  homogeneous polynomials. Thus, we get

$$\dim(\mathcal{K}') = \frac{(d-2)(d-1)}{2}.$$

On the other hand, by rank-nullity theorem, it follows:

$$\begin{aligned} \dim(\mathcal{K}') + \dim(\text{image}(\mathcal{M})) &= \frac{(d-2)(d-1)}{2} \\ \dim(\mathcal{K}) + \dim(\text{image}(\mathcal{M}^T)) &= 3d + l. \end{aligned}$$

Thus, since row rank and the column rank of a matrix are equal,

$$\dim(\mathcal{K}) = 3d + l - \frac{(d-2)(d-1)}{2} + \dim(\mathcal{K}') = l.$$

$\square$

**Corollary 3.3.** *Assume that  $\mathcal{M}$  has  $3n' + l$  rows, computed from the same number of points of the elliptic curve  $\mathcal{E}$ . If there is a curve  $\mathcal{C}$  intersecting  $\mathcal{E}$  non-trivially in  $3n'$  points among  $3n' + l$  points, then there is a vector  $v$  in  $\mathcal{K}$  with at least  $l$  zeros. Conversely, if there is a vector  $v$  in  $\mathcal{K}$  with at least  $l$  zeros, then there is a curve  $\mathcal{C}$  passing through those  $3n'$  points that correspond to the non-zero entries of  $v$  in  $\mathcal{M}$ .*

*Proof.* Assume that there is a non-trivial curve  $\mathcal{C}$  intersecting  $\mathcal{E}$  in  $3n'$  points. Then construct the matrix  $\mathcal{M}'$  whose rows are the points of intersection. Then from the earlier theorem we see that  $\mathcal{K}$  for this matrix  $\mathcal{M}'$  is non-zero. In all the vectors of  $\mathcal{K}$  if we put zeros in the place where we deleted rows then those are element of the left kernel of  $\mathcal{M}$ . It is clear that these vectors will have at least  $l$  zeros.

Conversely, if there is a vector with at least  $l$  zeros in  $\mathcal{K}$ , then by deleting  $l$  zeros from the vector and corresponding rows from  $\mathcal{M}$  we have the required result from the theorem above.  $\square$

In this paper we only deal with the case when there is exactly  $l$  zeros. This was we do not bother ourselves with intersection multiplicities.

## 4 Algorithms

Now we are ready for the algorithms. Recall that we have chosen a positive integer  $n'$  and  $k$  is such that  $k + 1 = 3n'$ .

### Main algorithm1:

**Step a:** Find  $k$  random distinct integers  $n_i$  and compute  $n_i P = P_i$  and then  $\overline{P_i}$ .

**Step b:** Form the matrix  $\mathcal{M}$  with rows  $\overline{P_i}$  and the last row  $-\overline{Q}$ .

**Step b:** Compute the kernel  $\mathcal{K}$  of  $\mathcal{M}$ .

**Step c:** If  $\mathcal{K}$  is non-zero STOP and output  $\sum_{i=1}^k n_i$ . Else go back to Step a.

### 4.1 What is the probability of success?

One can describe the above algorithm as an experiment. In that, we draw  $k$  points from  $p$  points without replacement. This can be done in  $\binom{p}{k}$  ways. Favorable number of events are the ones that sum upto  $m$ . So the number of favorable points is the number of unique partitions of  $m$  into  $k$  parts, denoted by  $q(m, k)$ . So the probability is

$$\frac{q(m, k)}{\binom{p}{k}}. \quad (5)$$

Knessl and Keller [4, Equation 3.9] shows that  $q(m, k) \sim \frac{m^{k-1}}{k[(k-1)!]^2}$ . Using their estimate the probability is approximately

$$\frac{m^{k-1}(p-k)!}{p!(k-1)!}. \quad (6)$$

Recall that  $m$  is unknown. We now study two ways how one can increase this probability.

## 4.2 How to make the probability bigger

In this section we increase the probability by introducing  $r$  more points of  $\mathcal{E}$  into the matrix  $\mathcal{M}$ . So the new algorithm is as follows:

**Main algorithm2:**

**Step a:** Find  $k+r$  random distinct integers  $n_i$  and compute  $n_i P = P_i$  and then  $\overline{P_i}$ .

**Step b:** Create the same matrix  $\mathcal{M}$  with  $\overline{P_i}$  and  $\overline{-Q}$  as before and compute its left-kernel  $\mathcal{K}$ .

**Step c:** If  $\mathcal{K}$  contains a vector  $v$  with exactly  $r$  zeros STOP and output  $\sum_{i \in T} n_i$ . Where  $T$  is a subset of  $\{1, 2, \dots, k+r\}$  where  $v$  has non-zero entries. Else go back to Step a.

It follows easily that the probability of success of this improved attack is

$$\binom{k+r}{k} \frac{m^{k-1}(p-k)!}{p!(k-1)!}. \quad (7)$$

## 4.3 How to make the probability estimate reliable

One of the big problem with this algorithm is that it depends on the discrete logarithm  $m$ , which by definition is unknown. The estimate of  $q(m, k)$  depends heavily on  $m$ . So the probability estimates depend on  $m$  so heavily that it will be useless for predicting the success of our algorithm. So we do what people in stock market does – spread the risk. Instead of only talking about  $Q = mP$ , we construct  $Q_i = m_i Q$ , where each  $m_i$  is randomly chosen between 1 and  $p$ .

The new algorithm is the following:

**Main algorithm3:**

**Step a:** Find  $k+r$  random distinct integers  $n_i$  and compute  $n_i P = P_i$  and then  $\overline{P_i}$ .

**Step b:** Find  $t$  random distinct integers  $m_i$  and compute  $m_i Q = Q_i$  and then  $\overline{Q_i}$ .

**Step c:** Form the matrix  $\mathcal{M}$  as follows: first  $k + r$  rows  $\overline{P_i}$  and then  $\overline{-Q}$  and the next  $t$  rows  $\overline{-Q_i}$ .

**Step d:** Form the matrix  $\mathcal{M}$  and compute its left-kernel  $\mathcal{K}$ .

**Step e:** If  $\mathcal{K}$  contains a vector  $v$  with exactly  $r$  zeros in  $\{1, 2, \dots, k + r\}$  and  $t$  zeros in  $\{k + r + 1, k + r + 2, \dots, k + r + t + 1\}$  STOP and output  $m_\gamma^{-1} \sum_{i \in T} n_i$  where  $\gamma$  is the index of the non-zero entry in  $\{k + r + 1, k + r + 2, \dots, k + r + t + 1\}$  minus  $k + r + 1$  with the understanding that  $m_0 = 1$  and  $T$  is a subset of  $\{1, 2, \dots, k + r\}$  where  $v$  has non-zero entries. Else go back to Step a.

**Note:** One can extend this algorithm for less than  $t$  zeros as well.

**Probability estimate:** Now we try to estimate the expected value of success of the above algorithm with  $r$  extra-points  $P_i$  and  $t$  extra-points  $Q_i$ . To estimate we use the approximation that  $q(m, k) \sim \frac{m^{k-1}}{s[(s-1)!]^2}$  from the work of Knessl and Keller [4, Equation 3.9]. We furthermore assume that  $m$  is uniformly distributed and by construction  $m_i$  are uniformly and independently distributed as well. Using the fact that if  $X_1, X_2, \dots, X_{t+1}$  are independent random variables, their expectation  $E(X_1 + X_2 + \dots + X_{t+1}) = E(X_1) + E(X_2) + \dots + E(X_{t+1})$ , it suffices to compute the expectation for any one  $Q_i$ . We will do that for  $Q$ .

From the earlier estimate, it is easy to see that the expectation for  $Q$  is

$$\binom{k+r}{k} \frac{(p-k)!}{p!(k-1)!} \sum_{m=1}^p m^k$$

and for  $t + 1$  random variable it is

$$\binom{k+r}{k} \frac{(t+1)(p-k)!}{p!(k-1)!} \sum_{m=1}^p m^k. \quad (8)$$

## 5 Implementation and complexity

We implemented many versions of the above three algorithms over the years in Sage [1]. We will only talk about the latest, and hopefully the better implementation. These implementations were done as a proof of concept and was not not optimized for speed.

First step of the process was to create the matrix  $\mathcal{M}$ . In this case we fixed  $n' = 25$ ,  $r = 5$  and  $t = 5$ . Then compute the left-kernel  $\mathcal{K}$  of  $\mathcal{M}$ .



In this case  $\mathcal{K}$  is actually a matrix for  $\mathcal{K}$  with a basis as rows. This was a reasonably fast process with complexity  $O(n'^6)$ .

Then the problem was to test if  $\mathcal{K}$  contains a vector of the required form. Recall that we are looking for a vector  $v$  with exactly  $r$ -zeros from the left  $k + r$  entries and then only one non-zero from the right  $t + 1$  entries. If there is such a  $v$ , then we have solved the discrete logarithm problem. The question is how to test, if there is such a  $v$ .

For this we used a well known theorem that rows of a square matrix are linearly dependent if and only if the determinant is zero. Recall that from Lemma 3.1, the dimension of  $\mathcal{K}$  is  $r + t$ . Then we picked all possible matrices with  $r$  columns from the left  $k + r$  columns of  $\mathcal{K}$  and  $t$  columns from the right  $t + 1$  columns. For each picking, we end up with square matrix of size  $(r + t) \times (r + t)$ . We check for the determinant and stop the algorithm if the determinant is zero.

The idea works because, if the determinant is zero, there is a linear combination of the basis of  $\mathcal{K}$  that has zero in the right place. We successfully tested this idea for many different field on many different sizes. However, we felt that computing the determinant for all possible sub matrices is a bottleneck for this algorithm and ways should be found to make this more efficient. The largest group of elliptic group of prime order that we tested is 129159847, in which a discrete logarithm problem was solved.

## References

- [1] The Sage Developers. *SageMath, the Sage Mathematics Software System*, 2016. <http://www.sagemath.org>.
- [2] William Fulton. *Algebraic Curves*. self-published, 2008.
- [3] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. *An introduction to mathematical cryptography*. Springer, 2008.
- [4] Charles Knessl and Joseph B. Keller. Partition asymptotics from recursion equations. *SIAM journal of applied mathematics*, 50(2):323–338, 1990.
- [5] J. S. Milne. *Elliptic Curves*. BookSurge Publishers, 2006.